DiffuSyn: A Diffusion-Driven Framework With Syntactic Dependency for Aspect Sentiment Triplet Extraction

Qiuhua Yi, Xiangjie Kong, *Senior Member, IEEE*, Linan Zhu, Chenwei Zhang, and Guojiang Shen

***Abstract*—Aspect Sentiment Triplet Extraction (ASTE) is a fine-grained sentiment analysis task that involves identifying aspect and opinion terms and conducting sentiment analysis for each aspect-opinion pair. We categorize existing methods into tagging-based, span-based, and generation methods. However, despite notable achievements, these methods still face certain challenges: span-based methods struggle to distinguish similar spans, and generation methods are prone to slower decoding speeds. To address these challenges, we propose a novel ASTE framework, called DiffuSyn, which leverages diffusion models and syntactic dependency parsing to improve the performance of sentiment analysis. Specifically, we first add Gaussian noise to the indices at the beginning and end of aspect words and opinion words in sentences based on a non-autoregressive diffusion model. We obtain accurate boundaries of aspect and opinion terms through boundary denoising, and thus identify precise spans. Secondly, we introduce a syntactic dependency parser to capture the syntactic dependencies within review sentences, thereby providing effective syntactic information for sentiment analysis. Finally, we complete the matching of aspects and opinions by leveraging the extracted boundaries and syntactic semantic information, facilitating the prediction of sentiment relationships. We conduct experiments on four public datasets (ASTE-Data-V2), and the results indicate the effectiveness of our approach in the ASTE task. Furthermore, our method achieves state-of-the-art performance in both Aspect Term Extraction (ATE) and Opinion Term Extraction (OTE) tasks.**

***Index Terms*—Aspect sentiment triplet extraction, boundary denoising, diffusion model.**

I. Introduction

ASPECT Sentiment Triplet Extraction (ASTE) is an emerging fine-grained sentiment analysis task within the domain of Aspect-Based Sentiment Analysis (ABSA), which aims to deeply mine the rich human sentiment information contained in sentences of review texts [1], [2]. The essence of this task, as illustrated in Fig. 1, is to identify aspect terms and opinion terms and perform fine-grained sentiment analysis for each aspect-opinion pair. In the example sentence "The fish soup is tasty, but the salad is disappointing.", the aspect terms are $fish\ soup$ and $salad$, and the opinion terms are $tasty$ and $disappointing$. Ultimately, the ASTE task is to derive two triplets from this sentence: $(fish\ soup, tasty, Positive)$ and $(salad, disappointing, Negative)$.

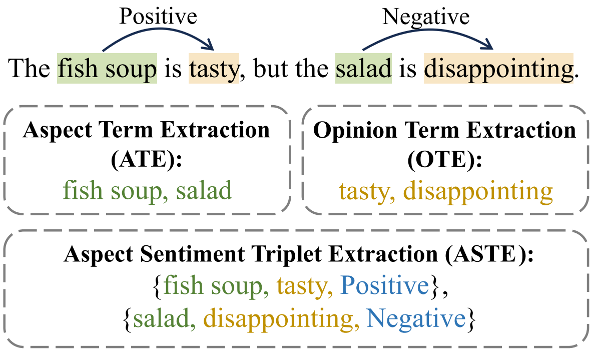
Positive
Negative
The fish soup is tasty, but the salad is disappointing.
Aspect Term Extraction (ATE):
fish soup, salad
Opinion Term Extraction (OTE):
tasty, disappointing
Aspect Sentiment Triplet Extraction (ASTE):
{fish soup, tasty, Positive},
{salad, disappointing, Negative}

Fig. 1. An example of Aspect Sentiment Triplet Extraction. The aspects, opinion expressions, and sentiments are marked with green, yellow, and blue, respectively.

ASTE is a multilevel task that involves not only discerning the target words and sentiment polarity but also delving into the human perspective on these target words. In other words, ASTE is concerned with identifying what the aspects are, determining the sentiment polarity associated with them, and understanding the reasons behind the emergence of such sentiment polarity. There exists a close correlation among these three sentiment elements. While aspect and opinion words typically present a one-to-one relationship, there are also one-to-many or many-to-one cases, which may introduce complexity and potential sentiment inconsistency [3]. Furthermore, the transmission of sentiment in opinion terms mainly relies on the accurate extraction of aspect-opinion pairs [4], so the correct acquisition of aspect-opinion pairs in the ASTE task is of paramount importance [5]. The complexity of ASTE and its requirement for deep contextual understanding make it a fascinating and challenging area of research.

Received 6 March 2024; revised 31 August 2024; accepted 19 January 2025. Date of publication 29 January 2025; date of current version 17 February 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62072409 and Grant 62176234, in part by the Zhejiang Provincial Natural Science Foundation under Grant LR21F020003, and in part by the Fundamental Research Funds for the Provincial Universities of Zhejiang under Grant RF-B2020001. The associate editor coordinating the review of this article and approving it for publication was Prof. Mark Hasegawa-Johnson. *(Corresponding author: Xiangjie Kong.)*

Qiuhua Yi, Xiangjie Kong, Linan Zhu, and Guojiang Shen are with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China, and also with the Zhejiang Key Laboratory of Visual Information Intelligent Processing, Hangzhou 310023, China (e-mail: qiuhuayi@outlook.com; xjkong@ieee.org; zln@zjut.edu.cn; gjshen1975@zjut.edu.cn).

Chenwei Zhang is with the School of Faculty of Education, University of Hong Kong,, Hong Kong (e-mail: chwzhang@hku.hk).

Digital Object Identifier 10.1109/TASLPRO.2025.3536179

In order to accurately extract the sentiment triplets, researchers have carried out a great deal of work, and we classify the current mainstream methods into three categories: tagging-based methods, span-based methods, and generation methods. Tagging-based methods, the earliest proposed approaches [6], are based on the construction of sequences or grids by annotating the relationships between aspect terms and opinion terms, along with sentiment polarity [4], [7], [8], [9]. Though easy to implement, they struggle with the complexity of one-to-many or many-to-one aspect-opinion pairs and cannot guarantee the consistency of sentiment. To overcome these challenges, a series of span-based approaches subsequently emerged [10], [11], [12], [13], [14]. These transform the problem into a multi-turn reading comprehension task by querying to obtain the corresponding spans [10], [11], or enumerate all possible spans in the sentence [12], [13], [14], obtaining the aspect and opinion spans through a series of filtering and pruning operations. However, these methods tend to cause redundancy and increase the computational overhead. A third distinct approach [15], [16], [17] proposes to treat the sentiment triplet extraction task as a sequence generation task. For instance, in [16], sentiment elements are aggregated into a sequence using templates, and the corresponding sentiment elements are extracted from the templates based on rules. However, generation-based methods typically rely on autoregressive models, resulting in a time-consuming generation process, especially when handling long sentences or large datasets, where slow decoding becomes a significant issue. Additionally, these models may introduce words from outside the sentence during generation, particularly when dealing with open-domain text, leading to the inclusion of words not present in the original input. This can interfere with the model's performance, resulting in inaccurate or irrelevant sentiment triplets.

Although existing methods have achieved remarkable success, they still face some challenges: (1) Span-based methods struggle to effectively distinguish similar spans, especially those with partial token overlap, such as "soup" and "fish soup". This overlap directly impacts the accurate identification of boundaries, resulting in conflicting spans and subsequently affecting the accuracy of sentiment triplets. (2) While generative methods leverage pre-trained language models, showcasing robust generation capabilities, decoding efficiency remains a concern. The relatively slow decoding speed during the generation process limits their widespread applicability in real-world scenarios.

In response to these challenges, we turn to an emerging generative model, the diffusion model, and propose a **Diffu**sion-Driven Framework with **Syn**tactic Dependency for Aspect Sentiment Triplet Extraction (**DiffuSyn**). The diffusion model, initially applied in the field of image generation, has demonstrated significant potential in NLP over the past two years, and its training involves two processes, known respectively as the forward and the reverse process. Our model is the inaugural attempt to apply the diffusion model to the ASTE task, and our motivation is to utilize the diffusion model to accurately identify the boundaries between aspect and opinion spans, thereby avoiding the generation of words outside the original sentence. Additionally, due to the non-autoregressive nature of the diffusion model, it also offers significant advantages in inference speed. Firstly, we initiate the process by applying the diffusion model to extract aspect and opinion terms, and the original data for the diffusion model are the starting and ending indices of aspect and opinion terms in the comment text. In the forward process, Gaussian noise is continually added to the data, while in the reverse process, denoising progressively identifies boundaries from the noisy spans. Secondly, we design a syntactic dependency analyzer, where we extract the syntactic dependencies and the distance representations of aspect-opinion pairs in sentences to capture long-distance dependencies in the text and provide effective syntactic information for aspect-based sentiment analysis. Finally, integrating semantic information from the text, we employ a sentiment classification model to accurately identify the sentiment and obtain the final sentiment triplets.

To test our model, we conduct extensive experiments on four publicly available datasets. Compared with nine baseline methods, the experimental results demonstrate the effectiveness of our approach in the ASTE task. Simultaneously, our model achieves state-of-the-art performance in ATE and OTE tasks. Our contributions can be summarized as follows:

- We propose a diffusion-based approach to extract aspects and opinions, which utilizes the denoising capability of diffusion models to stepwise extract the boundaries of spans, overcoming boundary uncertainty and reducing false candidate spans common in traditional methods. To the best of our knowledge, it is the first time that the diffusion model has been applied to an ASTE.
- We design a syntactic dependency analyzer that integrates sentence dependency relations with the distance representation of aspect-opinion pairs, aiming to address the issue of long-distance dependencies in ASTE tasks.
- We conduct extensive experiments on four publicly available datasets, and the results show that our approach achieves better results on the ASTE task while outperforming previous methods on the ATE and OTE tasks.

II. Related Work

A. Aspect Sentiment Triplet Extraction

Aspect-based sentiment analysis is a fine-grained sentiment analysis task aimed at recognizing human emotions at the aspect level. Depending on the number of output sentiment elements, aspect-based sentiment analysis is classified into single ABSA tasks and compound ABSA tasks [1]. Specifically, Aspect Term Extraction (ATE) [18], [19], [20] and Opinion Term Extraction (OTE) [19], [20], [21] are examples of single tasks, whereas Aspect-Opinion Pair Extraction (AOPE) [22], [23] is an example of a compound task. It is noteworthy that compound sentiment analysis tasks have received considerable attention lately, such as the Aspect Sentiment Triplet Extraction (ASTE) work that [6] proposed in 2020. We divide the several approaches that have been presented by later scholars for this task into three basic categories: tagging-based methods, span-based methods, and generation methods.

Tagging-based methods represent the most widely used approach in the early studies on aspect sentiment triplet extraction.

The first study of ASTE [6] proposed a unified tagging scheme known as BIOSE. This scheme, however, did not include any positional encoding information, which limited the annotations' expressiveness and led to a disconnected relationship between opinion and aspect. To address this limitation, [4] proposed an improved scheme that labels the locations of aspects and opinions and includes information about their relationship. Additionally, [7] introduced a method called the Grid Tagging Scheme, which tags the relationships between word pairs in a grid to accomplish aspect-oriented fine-grained opinion extraction. Similarly, [8] and [9] proposed using Table Filling, another tagging-based method. However, these tagging methods may lead to cascading errors, and the sentiment labels applied to each word may give rise to issues of inconsistent sentiment.

Span-based methods currently focus on two perspectives: Machine Reading Comprehension (MRC) [10], [11] and span classification [12], [13], [14]. Dual-MRC [10] and BMRC [11] transform ASTE into a multi-turn MRC task, where spans are obtained by designing various queries to establish connections between various subtasks. Sentiment classification is subsequently conducted based on the identified spans. Similarly, span classification involves initially extracting a candidate set for aspects and opinions, then matching the aspect and opinion pairs and performing sentiment prediction [12], [13], [14]. To elaborate, [12] utilized a sliding window to enumerate all possible spans, followed by filtering to retain effective aspect terms and opinion terms for pairing and sentiment categorization. Unfortunately, this approach may lead to redundancy in aspects and opinions, which would then cause the triplet redundancy issue.

Generation methods transform the triplet extraction task into a generation task, typically employing a pre-trained sequence-to-sequence model. [15] utilized the BART model to generate sequences, where the sequence comprises the index positions of the aspect and opinion, along with the sentiment polarity. On the other hand, [16] and [17] cast the ASTE task as a text generation process, generating target sentences based on designed templates and directly extracting triplets from the sentences. Despite achieving a certain degree of effectiveness, this generative method suffers from the inefficiency of the autoregressive decoding process during generation and is prone to generating words outside the scope of the review sentence.

In contrast to the above, our proposed DiffuSyn is the first method to apply a diffusion model for Aspect Sentiment Triplet Extraction, which is able to gradually refine the boundaries of both aspect and opinion. In contrast to span-based methods, DiffuSyn can minimize redundant spans. Furthermore, when compared to generation methods, the diffusion model can generate all spans in parallel in a non-autoregressive way, thereby enhancing the decoding efficiency.

B. Diffusion Model

The history of diffusion models dates back to 2015, when [24] introduced the concept of Diffusion Probabilistic Models (DPM). However, it was not until 2020, with the successive proposal of DDPM [25] and DDIM [26], that diffusion models began to attract widespread attention. In recent years, diffusion models have made substantial progress in the field of image and audio generation [27], [28], [29], [30], [31], demonstrating their great potential in generative applications. The diffusion model consists of two processes: the forward process continuously introduces noise to destroy the original data, while the reverse process uses denoising to recover the data.

Although diffusion model have achieved tremendous success in continuous spaces such as images, their application to discrete data still faces various challenges. Some works have successfully applied the diffusion model to discrete text [32], which can be categorized into two approaches: one involves mapping discrete text to a continuous space [33], [34], [35], and the other generalizes the diffusion model to discrete text [36], [37]. However, both of these methods rely on the forward noise addition and reverse denoising processes. To transform discrete tokens into continuous embeddings, Diffusion-LM [33] adds embedding and rounding phases at the start of the forward process and the finish of the reverse process. The intermediate process is not much different from the standard continuous diffusion model, involving the addition of Gaussian noise. DiffusER [36], on the other hand, conducts a series of diffusion steps at the token level, considering the editing operations on tokens as a kind of noise.

As in continuous diffusion models, our method encodes the boundary information of aspects and opinions into a continuous space for diffusion steps, thereby achieving the extraction of discrete text.

III. Preliminaries

Both forward and reverse stages of the diffusion process can be regarded as parameterized Markov chains. The forward process involves gradually adding Gaussian noise to the original data, progressing until the data transforms into pure Gaussian noise. Given the original data $x_0 \sim q(x_0)$, each step of noise addition is performed on the basis of the preceding data x_{t-1} in the following manner:

$$q(\mathrm{x}_t|\mathrm{x}_{t-1}) = \mathcal{N}(\mathrm{x}_t; \sqrt{1-\beta}_t\mathrm{x}_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

where β_t represents the variance schedule adopted at each step, with values ranging between 0 and 1. With successive steps, the value of β_t gradually escalates. Each step of the diffusion process generates data x_t with added noise, making the whole diffusion process a Markov chain:

$$q(\mathrm{x}_{1:T}|\mathrm{x}_0) = \prod_{t=1}^{T} q(\mathrm{x}_t|\mathrm{x}_{t-1}) \tag{2}$$

In practice, since the noise added at each step is independent and normally distributed, we are able to obtain the noisy result x_t at any timestep t directly from the original data x_0. Given $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, through a series of derivations, we can obtain:

$$q(\mathrm{x}_t|\mathrm{x}_0) = \mathcal{N}\left(\mathrm{x}_t; \sqrt{\bar{\alpha}_t}\mathrm{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right) \tag{3}$$

The reverse process is a denoising procedure. If we have knowledge of the true distribution at each step of the reverse process, denoted $q(x_{t-1}|x_t)$, starting from a random noise x_T,

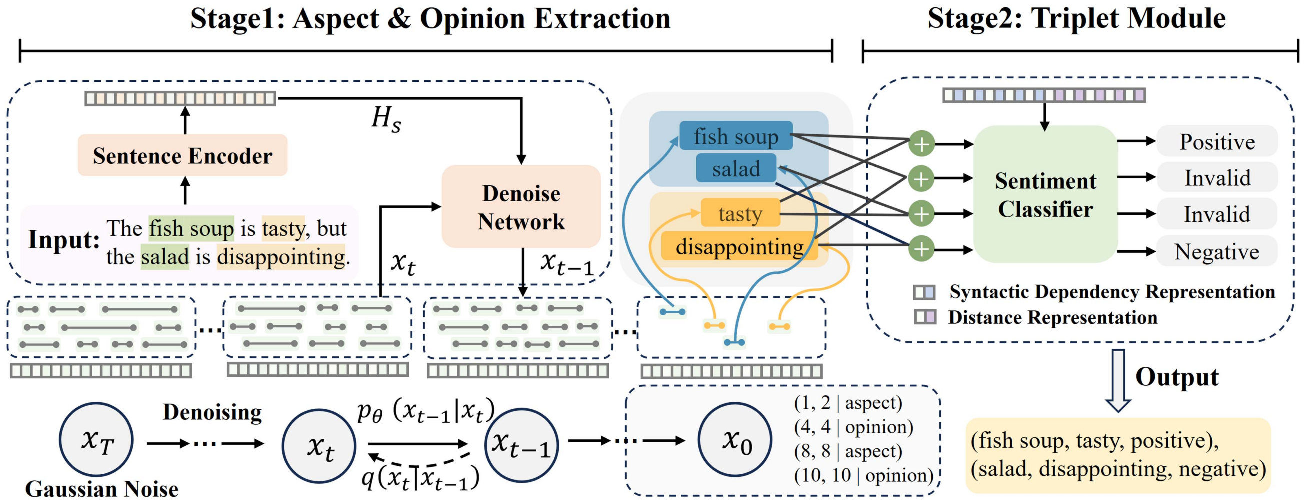

Fig. 2. Overview of DiffuSyn. Stage 1 describes the process of boundary denoising diffusion to extract aspects and opinions, and Stage 2 delineates the pairing of aspects and opinions along with the prediction of sentiment polarity.

gradually denoising allows us to obtain a real sample x_0. To estimate the distribution $q(x_{t-1}|x_t)$, we can employ a neural network, defining the reverse process as another Markov chain:

$$p_\theta(\mathrm{x}_{0:T}) = p(\mathrm{x}_T)\prod_{t=1}^{T} p_\theta(\mathrm{x}_{t-1}|\mathrm{x}_t) \tag{4}$$

$$p_\theta(\mathrm{x}_{t-1}|\mathrm{x}_t) = \mathcal{N}(\mathrm{x}_{t-1}; \mu_\theta(\mathrm{x}_t, t), \Sigma_\theta(\mathrm{x}_t, t)) \tag{5}$$

where $p(x_T)$ is the standard normal distribution $N(x_T; 0, I)$, and $p_\theta(x_{t-1}|x_t)$ is a parameterized Gaussian distribution. θ represents the parameter in the learnable network model, and $\mu_\theta(x_t|t)$ is the model's prediction for the variance and mean of $q(x_{t-1}|x_t)$. We adopt a fixed variance $\Sigma_\theta(x_t|t) = \sigma_t^2 I$, requiring only the prediction of the mean $\mu_\theta(x_t|t)$. Thus, we have:

$$\begin{aligned} q(\mathrm{x}_{t-1}|\mathrm{x}_t, \mathrm{x}_0) &= \mathcal{N}(\mathrm{x}_{t-1}; \tilde{\mu}(\mathrm{x}_t, \mathrm{x}_0), \sigma_t^2\mathbf{I}) p_\theta(\mathrm{x}_{t-1}|\mathrm{x}_t) \\ &= \mathcal{N}(\mathrm{x}_{t-1}; \mu(\mathrm{x}_t, t), \sigma_t^2\mathbf{I}) \end{aligned} \tag{6}$$

The entire loss function is $-\log(p_\theta(x_0))$, where the objective is to maximize the probability of generating x_0 by adjusting the parameters θ of the neural network. Ultimately, this problem is transformed into calculating a variational lower bound for this objective, and obtaining a more computationally tractable formulation:

$$L_s = \mathbb{E}_{\mathrm{x}_0, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})}\left[\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}\mathrm{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|^2\right] \tag{7}$$

IV. Methodology

A. Task Definition

Consider a sentence $S = \{w_1, w_2, \ldots, w_n\}$, where w_i denotes a token in the sentence and n denotes the length of the sentence. The goal of the ASTE task is to extract a set of sentiment triples $\mathcal{P} = \{(a, o, c)_k\}_{k=1}^{|\mathcal{P}|}$ from the sentence S, where a, o, and c represent the aspect term, opinion term, and sentiment polarity, respectively, with $c \in \{Positive, Neutral, Negative\}$. Aspect and opinion terms can be represented by a set $E = \{(l, r, t)_m\}_{m=1}^{|E|}$, where l and r denote the boundary indices of the aspect and opinion terms, and t represents the type with $t \in \{Aspect, Opinion\}$.

B. Boundary Denoising Diffusion Strategy

We regard the extraction of aspects and opinions to be a boundary denoising diffusion process, as shown in Fig. 2. Within the diffusion process, the start and end boundaries of aspects and opinions are viewed as data samples. During the forward process, we continuously add Gaussian noise to the boundary data; in the reverse process, we gradually denoise to refine the exact boundaries of aspects and opinions, ultimately achieving accurate extraction of these terms. This boundary denoising diffusion strategy provides an effective framework for exploring aspect and opinion information in textual data.

1) Forward Diffusion: The forward process is an iterative process that continuously adds Gaussian noise to the boundary data. Considering the possible variance in the number of aspects and opinions in different samples, we use a padding strategy to ensure the shape consistency of the data samples. Specifically, we expand the total number of aspects and opinions to a fixed length N. The expanded data is represented as $\mathbf{X} \in \mathbb{R}^{N\times 2}$, where two dimensions correspond to the start and end boundaries. For the padding method, we tried two approaches: random padding and repeated padding. Random padding samples the noise at random from a Gaussian distribution, while repeated padding replicates the true spanning noise. These padding strategies are designed to overcome inconsistencies due to initial data discrepancies, ensuring that the shape and structure of the data remain consistent during noise processing.

Given a data sample $\mathrm{x}_0 = \mathbf{X}$, according to the forward diffusion process, for each forward timestep $t \in [1, 2, \ldots, T]$, we can directly obtain the noise span at timestep t as follows:

$$\mathrm{x}_t = \sqrt{\bar{\alpha}}\mathrm{x}_0 + \sqrt{1-\bar{\alpha}}\epsilon \tag{8}$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is noise sampled from the standard Gaussian distribution. When Gaussian noise is added gradually, x_T will tend toward a fully Gaussian noise, i.e., $\mathrm{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

2) Reverse Diffusion: The reverse process is a boundary denoising procedure that begins at x_T and generates x_0 through a well-trained model. Here, we employ the DDIM algorithm to implement the reverse process, which exhibits a faster generation speed. We sample a subsequence $[\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_\sigma]$ of length σ from the original timesteps $[1, 2, \ldots, \mathcal{T}]$, where $\sigma < \mathcal{T}$, to reduce the number of generated iterations, thereby improving generation efficiency. Similarly, defining the forward process of $\mathrm{x}_{\mathcal{T}_1}, \mathrm{x}_{\mathcal{T}_2}, \ldots, \mathrm{x}_{\mathcal{T}_\sigma}$ as a Markov chain, the generation process can be replaced by the reverse Markov chain of this subsequence. The generation process of DDIM does not introduce random noise; it is a deterministic procedure. Based on the form of $q(\mathrm{x}_{t-1}|\mathrm{x}_t, \mathrm{x}_0)$, the generation of x_{t-1} from x_t during the generation phase can be expressed using the following formula:

$$\begin{aligned} \mathrm{x}_{\mathcal{T}_{i-1}} &= \sqrt{\alpha_{\mathcal{T}_{i-1}}} \left(\frac{\mathrm{x}_{\mathcal{T}_i} - \sqrt{1-\alpha_{\mathcal{T}_i}} \epsilon_\theta(\mathrm{x}_{\mathcal{T}_i}, \mathcal{T}_i)}{\alpha_{\mathcal{T}_i}} \right) \\ &\quad + \sqrt{1-\alpha_{\mathcal{T}_{i-1}}} \epsilon_\theta(\mathrm{x}_{\mathcal{T}_i}, \mathcal{T}_i) \\ &= \sqrt{\alpha_{\mathcal{T}_{i-1}}} \hat{\mathrm{x}}_0 + \sqrt{1-\alpha_{\mathcal{T}_{i-1}}} \epsilon_\theta(\mathrm{x}_{\mathcal{T}_i}, \mathcal{T}_i) \end{aligned} \tag{9}$$

where $\hat{\mathrm{x}}_0 = f_\theta(\mathrm{x}_{\mathcal{T}_i}, T_i, S)$ represents the predicted boundaries of aspect and opinion. Here, $f_\theta(\mathrm{x}_{\mathcal{T}_i}, \mathcal{T}_i, S)$ is a learnable neural network, which will be detailed in next subsection as the denoising network. $\mathrm{x}_{\mathcal{T}_i}$ is the noise sampled from a Gaussian distribution at timestep $\mathcal{T}_i$, and S represents the input sentence.

C. Denoising Network

Denoising network $f_\theta(\mathrm{x}_t, t, S)$ is a crucial component of the entire diffusion model, whose inputs are the noise span x_t at timestep t and the sentence S, and whose outputs are the predicted boundaries. The denoising network consists of three components: the Sentence Encoder, the Boundary Predictor and the Aspect and Opinion Classifier.

The Sentence Encoder is designed to transform the input sentence S into a sentence encoding H_S, utilizing BERT as the base model. Specifically, we input a sequence of words of sentence S into BERT, then extract a fixed-length vector as sentence encoding H_S from the last output layer of BERT, which captures the semantic information of sentence S. During the training process, in order to improve efficiency, we execute the sentence encoder only once for each sentence, meaning BERT is invoked just once, and then cache the sentence encoding H_S for subsequent use. Such a strategy helps optimize the training efficiency and reduce the computational cost.

Boundary prediction aims to refine the boundaries of aspects and opinions during the denoising process. Initially, we obtain the span representation H_X from the sentence encoding. To capture semantic information more comprehensively within the span representation, we introduce self-attention and cross-attention mechanisms. Specifically, self-attention is applied to H_X to obtain a new representation H'_X. Subsequently, we incorporate a cross-attention mechanism between H_S and H'_X, where H_S serves as both key and value, and H'_X as the query. The mathematical formulation for this design is as follows:

$$H'_X = \text{SelfAttention}(H_X) \tag{10}$$

$$H_{X_S} = \text{CrossAttention}(H_S, H'_X) \tag{11}$$

We use the sinusoidal positional embedding [25] to encode our timestep t, and the updated spanning representation can be expressed by the following formula:

$$H'_{X_S} = H_{X_S} + H_t \tag{12}$$

where H_t denotes the sinusoidal embedding at timestep t.

We construct two Boundary Predictors to precisely determine the left (l) and right (r) boundaries of aspect and opinion words. Specifically, we first calculate the fused representation of noise span and sentence representation, denoted $H^\lambda_{SX} = H_S W^\lambda_S + \hat{H}_{X_s} W^\lambda_{X_s}$, where $\lambda \in \{l, r\}$, with l indicating the left boundary and r indicating the right boundary. W^λ_S and $W^\lambda_{X_s}$ are two learnable matrices. The probabilities for the left and right boundaries of the words are:

$$P^\lambda = \text{sigmoid}(\text{MLP}(H^\lambda_{SX})) \tag{13}$$

where MLP represents a two-layer perceptron.

In order to compute $\hat{x}_0$, we obtain the indices of the boundaries for N noise spans through probability. If the current step is not the final denoising step, we iterate the next round of the backward diffusion process according to (9).

The Aspect and Opinion Classifier is employed to categorize the identified boundary spans into $aspect$, $opinion$, and $None$, where $None$ represents words other than aspect and opinion. This classifier can be represented by the following formula:

$$P^z = \text{sigmoid}(\text{MLP}(H'_{X_s})) \tag{14}$$

where MLP represents a two-layer perceptron.

We match the predicted aspect and opinion terms with the gold aspect and opinion terms to calculate the training loss. As in DiffusionNER [38], we initially use the Hungarian algorithm to find the optimal matching $\hat{M}$ between the predicted and true sets. We train by maximizing the probability of matching:

$$\mathcal{L}_{a,o} = -\sum_{i=1}^{N} \sum_{\lambda \in \{l,r,z\}} \log P^\lambda_i(\hat{M}^\lambda(i)) \tag{15}$$

where $\hat{M}^l(i)$, $\hat{M}^r(i)$, and $\hat{M}^z(i)$ respectively denote the indices and types of the left and right boundaries.

D. Inference

In the inference process, we first sample N noise spans from a Gaussian distribution; then, based on the timestep subsequence T of length σ, we iterate through denoising using a learned reverse denoising network. The left and right boundaries, as well as the type set $(l_i, r_i, z_i)^N_{i=1}$, are obtained according to (13) and (14), where $\lambda_i = argmax P^\lambda_i, \lambda \in \{l, r, z\}$. Considering the redundancy of the candidate set obtained in this way, we design a pruning operation. Specifically, for spans with identical left and right boundaries, only the span with the highest probability for the associated type is retained. In addition, we set a threshold for the sum of boundary and type probabilities, and if the sum

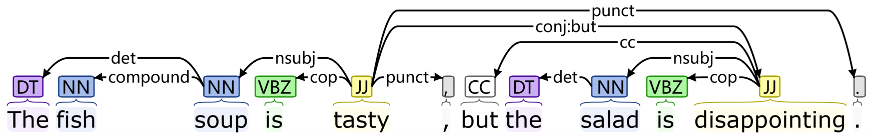

Fig. 3. An example of syntactic dependency analysis.[2]

of probabilities is lower than this threshold, we directly discard the candidate spans. Through this pruning process, we extract the aspect and opinion terms, which are the first two elements of the sentiment triplet.

E. Pairing

1) Aspect Opinion Pair Representation: We first use BERT to obtain the representation $\{h_1, h_2, \ldots, h_n\}$ for each token, and we define the representation $h_{start,end}$ for each span as

$$h_{start,end} = [h_{start}; h_{end}; \psi_{width}(start, end)] \tag{16}$$

where $start$ and end refer to the start and end indices of the aspect and opinion spans obtained from the diffusion model. $\psi_{width}(start, end)$ is a trainable feature embedding used to represent the span width.

2) Aspect Opinion Distance Representation: Given that in most cases, matching aspects and opinions are proximate in position, with the probability of valid aspect-opinion pairs formed by widely distant spans being relatively low, we introduce the following representations to better capture the positional information of aspects and opinions:

$$h_{distance} = \psi_{distance}(start^a, end^a, start^o, end^o) \tag{17}$$

where $\psi_{distance}$ is a trainable feature embedding.

3) Syntactic Dependency Representation: In the task of matching aspects and opinions, apart from positional information, the syntactic dependency of the sentence also play a crucial role. Fig. 3 illustrates the syntactic dependency for the sentence "The fish soup is tasty, but the salad is disappointing". In this diagram, $fish$ and $soup$ are connected by the $compound$ relation, indicating that they form a composite structure in which $fish$ is a modifier of $soup$. In the dependent syntactic tree, $fish\ soup$ and $salad$ are noun subjects ($nsubj$), is connects $fish\ soup$ and $tasty$ through the cop relation, and $salad$ and $disappointing$ are the same, representing the connection between them. This indicates that $tasty$ and $disappointing$ are attributes associated with the preceding nouns $fish\ soup$ and $salad$. In our work, we introduce a syntactic dependency representation with the following formula:

$$h_{dependency} = \psi_{dependency}(start^a, end^a, start^o, end^o) \tag{18}$$

where $\psi_{dependency}$ represents a feature embedding. All dependency parsing relationships in the sentences are implemented through SpaCy.[1]

To form a comprehensive and robust representation of aspect-opinion pairs, we integrate semantic representation, aspect-opinion distance representation, and syntactic dependency representation. This integrated approach effectively captures both positional and syntactic information, enhancing the model's ability to accurately relate aspects and opinions in the ASTE task. The integrated representation is defined as follows:

$$H_{a,o} = [h_{pair}; h_{distance}; h_{dependency}] \tag{19}$$

F. Sentiment Classification

Sentiment Classifier is used to identify valid aspect-opinion pairs from N^2 pairs and predict the sentiment relation for each span pair. We feed the pair representation into a feedforward neural network, which outputs the probability of sentiment relationship c for each span pair. Specifically, the sentiment classifier can be expressed by the following formula:

$$P(c \mid H_{a,o}) = \text{softmax}(FFNN_c(H_{a,o})) \tag{20}$$

where $c \in \{Positive, Neutral, Negative, Invalid\}$, and $Invalid$ denotes that the aspect and opinion terms constitute an invalid match.

Our training target is the sum of negative likelihood logarithms, and the mathematical representation of the loss is as follows:

$$\mathcal{L}_R = - \sum_{h_{a,o} \in H_{a,o}} \log P(c^* | h_{a,o}) \tag{21}$$

where c^* is the true sentiment relationship for the aspect-opinion pair, and $H_{a,o}$ represents the representation for all aspect-opinion pairs.

V. Experimental Settings

A. Datasets

To validate the effectiveness of our proposed DiffuSyn framework, we conduct experiments on the ASTE-Data-V2[3] dataset covering the restaurant and laptop domains. The ASTE-Data-V2 dataset comprises four benchmark sub-datasets derived from three ABSA SemEval Challenges, namely SemEval-2014 [39], 2015 [40], and 2016 [41]. Each sub-dataset includes training, validation, and test sets, with each data instance consisting of a sentence paired with a corresponding list of aspect-opinion-sentiment triplets. As the original data from SemEval Challenges only included aspect terms and their associated sentiment elements, making it unsuitable for the direct extraction of aspect sentiment triplets, [6] released the initial version,

[1] https://spacy.io/models/en

[2] https://corenlp.run/

[3] https://github.com/xuuuluuu/SemEval-Triplet-data/tree/master/ASTE-Data-V2-EMNLP2020

TABLE I
STATISTICS OF THE DATASETS

Datasets		#S	POS	NEU	NEG	#SW	#MW
Lap14	Train	906	817	126	517	824	636
	Dev	219	169	36	141	190	156
	Test	328	364	63	116	291	252
Rest14	Train	1266	1692	166	480	1586	752
	Dev	310	404	54	119	388	189
	Test	492	773	66	155	657	337
Rest15	Train	605	783	25	205	678	335
	Dev	148	185	11	53	165	84
	Test	322	317	25	143	297	188
Rest16	Train	857	1015	50	329	918	476
	Dev	210	252	11	76	216	123
	Test	326	407	29	78	344	170

ASTE-Data-V1,[4] after annotating opinion terms as per [42]. Subsequently, [4] released the latest version, ASTE-Data-V2, after further optimizations, including the removal of conflicting sentiment triplets. The statistical details of ASTE-Data-V2 are presented in Table I. '#S' denotes the number of sentences, 'POS', 'NEU', and 'NEG' represent the numbers of positive, neutral, and negative triplets, respectively. '#SW' ('single word') refers to the number of triplets where both ATs (Aspect Terms) and OTs (Opinion Terms) are single-word spans; '#MW' ('multi-word') indicates the number of triplets where at least one term is a multi-word span.

B. Evaluation Metrics

Following earlier research, we assess the effectiveness of the various approaches by calculating precision (P), recall (R), and F1 score (F1) from the experimental data. We define a triplet as correct in the ASTE task if and only if its sentiment polarity, aspect span, and opinion span are completely consistent with the ground truth labels. Furthermore, in the first stage of the experiment, we take into account two subtasks of the ASTE task: aspect term extraction (ATE) and opinion term extraction (OTE). Regarding these two subtasks, an opinion or aspect term is deemed accurate only if its bounds coincide with the ground truth spans. In our experiments, we select the test results in which the model achieves optimal performance on the validation set.

C. Baselines

We compare our proposed model with three categories of baselines: tagging-based methods, span-based methods, generation methods and LLM-based methods. We utilize the ASTE-Data-V2 dataset, employing the following baselines to evaluate the ATE, OTE, and ASTE tasks.

1) Tagging-Based Methods:

- **Pipeline** [6], in the first stage, employs a unified BIOES tagging scheme for the extraction of aspect terms with sentiment polarity and opinion terms; the second stage then completes their matching, facilitating the extraction of triplets.
- **JET-BERT** [4] proposes a novel position-aware tagging scheme that incorporates the positional relationships of aspects and opinions in the tagging information to capture the interactions between elements to jointly extract triples.
- **HIM** [5] introduces a novel boundary-aware tagging scheme that captures relationships between different subtasks through hierarchical interactions, transforming the ASTE task into a boundary-words relation classification problem.

2) Span-Based Methods:

- **BMRC** [11] transforms the ASTE task into a multi-turn machine reading comprehension problem by designing three types of queries to obtain aspect spans, opinion spans, and sentiment polarity from two directions.
- **Span-ASTE** [12] employs a dual-channel span pruning strategy to filter effective aspect and opinion candidate sets from all enumerated spans, followed by further filtering to eliminate invalid triplets.
- **EMC-GCN** [9] defines ten relations for ASTE and utilizes a biaffine attention module to embed these relations into a table, from which triplets are extracted.
- **Span-GCN** [43] designs a graph convolutional network to emphasize key inter-word dependencies, which exploits contextual semantics and latent dependencies to improve the interactions between sentiment entities.

3) Generation Methods:

- **BARTABSA** [15] utilizes the pre-trained BART model to generate the starting and ending indices of aspects and opinions, along with their corresponding sentiment polarities, in an autoregressive manner.
- **GAS** [17] designs two paradigms, namely annotation-style and extraction-style, to transform ABSA into a unified generative task, while also incorporating a prediction normalization technique to optimize the generated output.

4) LLM-Based Methods:

- **GPT-3.5-turbo and GPT-4o [44]:** In recent years, large language models (LLMs) have made significant advancements in the field of natural language processing (NLP). We select GPT-3.5-turbo and GPT-4o, two prominent LLMs, to compare with our method under zero-shot and few-shots learning settings.

D. Implementation Details

We use HuggingFace's Transformers library to implement our model, choosing bert-large-cased as the base encoder. In all experiments, we choose Adam as the optimizer. During the aspect and opinion extraction phases, we conduct 100 epochs of training with a learning rate set at 4e-5 and a weight decay of 0.01. For boundary diffusion, we set the noise span $N = 60$, the timestep $t = 6000$, and the probability threshold $\epsilon = 2.5$. As for pairing and predicting sentiment polarity, we conduct 20 epochs of training with a learning rate of 3e-5 and a weight decay of 0.05. We select the optimal model weights based on the F1 score

[4] https://github.com/xuuuluuu/SemEval-Triplet-data/tree/master/ASTE-Data-V1-AAAI2020

TABLE II
EXPERIMENTAL RESULTS FOR ASTE TASKS †

	Model	Lap14			Rest14			Rest15			Rest16		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
Tagging-based	P-pipeline	40.40	47.24	43.50	44.18	62.99	51.89	40.97	54.68	46.79	46.76	62.97	53.62
	JET-BERT	55.39	47.33	51.04	70.56	55.94	62.40	64.45	51.96	57.53	70.42	58.37	63.83
	HIM	65.99	56.05	60.59	76.99	70.46	73.57	69.65	63.23	66.26	73.11	71.05	72.06
Span-based	BMRC	65.12	54.41	59.27	71.32	70.09	70.69	63.71	58.63	61.05	67.74	68.56	68.13
	Span-ASTE	63.44	55.84	59.38	72.89	70.89	71.85	62.18	64.45	63.27	69.45	71.17	70.26
	EMC-GCN	61.70	56.26	58.81	71.21	72.39	71.78	61.54	62.47	61.93	65.62	71.30	68.33
	SpanGCN	72.37	52.96	61.16	77.99	68.69	73.04	64.33	69.68	**66.9**	72.50	70.30	71.38
Generation	BARTABSA	61.41	56.19	58.69	65.52	64.99	65.25	59.14	59.38	59.26	66.6	68.68	67.62
	GAS	62.42	58.78	60.55	71.74	72.58	72.15	60.40	64.87	62.55	68.84	72.18	70.47
LLM-based	GPT-3.5-turbo zero-shot	30.04	41.04	34.69	44.88	55.13	49.48	36.02	53.40	43.02	39.92	57.78	47.22
	GPT-3.5-turbo few-shots	39.79	50.09	44.35	51.51	65.19	57.55	43.34	63.09	51.39	51.12	71.01	59.45
	GPT-4o zero-shot	17.81	22.55	19.90	32.99	38.13	35.37	27.85	37.73	32.05	32.17	43.00	36.80
	GPT-4o few-shots	38.23	48.61	42.80	54.11	66.20	59.55	45.57	60.41	51.95	52.90	71.01	60.63
	Ours	64.31	58.75	**61.41**	74.77	72.74	**73.74**	64.15	64.95	64.55	72.20	72.76	**72.48**

† State-of-the-art results are marked bold and the best baselines are underlined.

on the validation set and execute all the experiments in the study on a Linux server with GeForce RTX 3090.

VI. Results and Analysis

A. Main Results

Table II shows the Precision (%), Recall (%) and F1 score (%) of our proposed DiffuSyn on the ASTE-Data-V2 dataset against the nine baselines. The F1 scores of our framework exceed the best baseline by 0.25%, 0.17%, and 0.42% on three datasets, Lap14, Rest14, and Rest16, respectively, which fully demonstrates the effectiveness of our proposed framework. Furthermore, except for HIM, we observe that the performance of tagging-based methods is inferior to span-based and generative methods. HIM outperforms other tagging-based methods significantly, attributed to its utilization of deep interaction between tasks, mitigating the occurrence of tagging errors. Among span-based methods, most exhibit favorable performance, and in generative methods, GAS outperforms BARTABSA. Notably, our method outperforms the generative methods on all four datasets, with an average improvement of 1.615% in F1 score compared to GAS. These enhancements strongly indicate the effectiveness of utilizing boundary diffusion for identifying aspects and opinions in review sentences.

In addition, the comparison with LLM-based methods indicates that, although GPT-3.5 and GPT-4 possess strong generative capabilities, their performance in the ASTE task still has certain limitations. Compared to specifically designed models, the performance of LLMs is somewhat lacking, and even in few-shot learning scenarios, they have not achieved optimal results in the ASTE task.

In order to validate the effectiveness of our model, we use the benchmark dataset ASTE-Data-V2 on the ATE and OTE tasks and compare it with the models Span-ASTE, BMRC, and HIM, with results shown in Table III. Overall, our model achieved optimal F1 score on all datasets for both tasks, with an average improvement of 4.34% for ATE and 2.64% for OTE. It is worth noting that the significant performance improvement of our method is primarily attributed to the application of the diffusion model. By utilizing boundary diffusion, we are able to more precisely refine the boundaries of spans, which effectively identifies the boundaries of multi-word aspect and opinion terms. This approach avoids the redundancy issues present in traditional models and reduces the reliance on complex interactions between aspects and opinions, directly enhancing the model's performance. In contrast, Span-ASTE, by enumerating all spans, increases the likelihood of redundant spans, and the model relies on the interactions between aspects and opinions, thus making it difficult to accurately identify boundaries for multi-word aspects and opinion terms. In addition, we observe that although BMRC has an advantage in precision, its performance is subpar in terms of recall, resulting in poor overall performance. HIM introduces

TABLE III
EXPERIMENTAL RESULTS FOR ATE AND OTE TASKS

Dataset	Model	ATE			OTE		
		P	R	F1	P	R	F1
Lap14	Span-ASTE	81.48	86.39	83.86	83.00	82.28	82.63
	BMRC	88.89	67.39	76.66	87.11	64.14	73.88
	HIM	82.79	87.65	85.14	79.46	83.29	81.29
	Ours	85.74	88.34	**87.02**	83.51	84.04	**83.78**
Rest14	Span-ASTE	83.56	87.59	85.50	82.93	89.67	86.16
	BMRC	88.18	74.76	80.92	89.71	78.57	83.77
	HIM	82.87	89.42	86.01	84.99	90.05	87.44
	Ours	85.71	88.44	**87.06**	85.63	91.72	**88.57**
Rest15	Span-ASTE	78.97	84.68	81.72	77.36	84. 86	80.93
	BMRC	83.39	61.57	70.84	86.40	66.16	74.94
	HIM	81.72	84.28	82.96	78.21	83.18	80.62
	Ours	83.11	86.57	**84.81**	81.05	84.43	**82.71**
Rest16	Span-ASTE	79.78	88.50	83.89	82.59	90.91	86.54
	BMRC	87.38	78.10	81.29	88.25	80.63	83.05
	HIM	80.34	90.50	85.11	83.60	90.24	86.79
	Ours	83.64	89.38	**86.42**	83.24	90.85	**86.88**

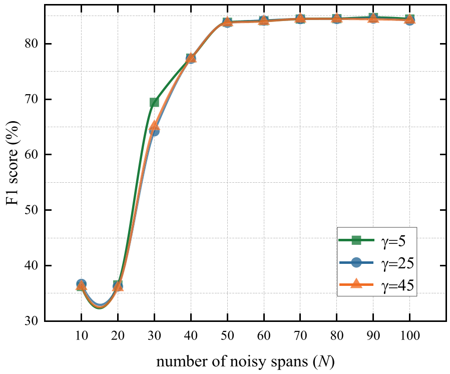

Fig. 4. Analysis of sampling noisy spans on Lap14.

a boundary-aware tagging scheme and shows significant results in experiments, but it requires extensive exploration of deep interactive information between subtasks. Our method identifies the boundaries of aspects and opinions by progressively denoising, demonstrating significant performance improvements and proving the effectiveness of boundary diffusion technology in aspect and opinion extraction.

B. Analysis

1) Numbers of Noisy Spans: During evaluation, our model can dynamically sample noise spans, meaning that we can sample different quantities of noise during testing. To gain a deeper understanding of the impact of sampling different noise spans on the research results, we conduct training with $N = 60$ noise spans on the Lap14 dataset. We evaluate the results using different sampling noise spans as shown in Fig. 4. We observe that the model's performance is significantly improved with increasing noise span and tends to a stable state. Specifically, when the noise span N is less than 50, the model's performance is relatively poor. This is due to the fact that the model is unable to cover all possible aspects and opinion terms with a small span. However, when $N >= 50$, our model performance starts to stabilize and becomes more reliable. Additionally, we conduct experiments with the sampling timestep γ set to 5, 25, and 45. The sampling timestep refers to the number of timesteps during the inference phase of the diffusion model, from the generated noise to the target data. We notice that the sampling timestep has a relatively small effect on the model performance. In our experiments, the model's performance remains relatively stable with variations in the sampling timestep, indicating a certain level of robustness in our model regarding the choice of sampling timestep.

2) Sizes of Probability Threshold: To deeply investigate the impact of setting the size of the probability threshold on the performance of the model when extracting aspects and opinions, we conduct an exhaustive evaluation of the results for different thresholds, whose results are depicted in Fig. 5. The observed trend is that, in both ATE and OTE tasks, the model's performance starts to decline when the threshold exceeds 2.7. We speculate that setting too large a threshold may filter out correctly predicted segments, thereby reducing the accuracy of the model. Therefore, choosing an appropriate threshold is crucial for improving results. In our work, we set a uniform threshold of 2.5 across datasets after comprehensive consideration.

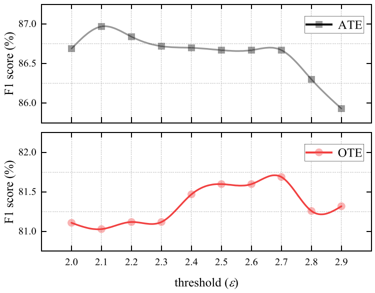

Fig. 5. Analysis of probability threshold on Lap14.

TABLE IV
Comparison With GAS on Inference Time for Dataset Lap14

Model	Inference Time(s)	Speed Comparison
GAS	32.47	1×
Ours (γ=1)	2.42	+13.42×
Ours (γ=5)	4.30	+7.56×
Ours (γ=10)	5.97	+5.44×
Ours (γ=20)	12.14	+2.67×
Ours (γ=40)	16.52	+1.96×
Ours (γ=80)	35.38	-0.92×
Ours (γ=160)	62.26	-0.52×

3) Inference Efficiency: To further verify the advantage of our method in inference efficiency, we conduct comparative experiments with another generation-based method, GAS [17]. As shown in Table IV, our model performs the best in terms of inference time. When the denoising timestep τ, which represents the number of sampling timestep from generated noise to target data in the inference phase, is set to 1, our model's inference speed is 13 times faster than GAS. Even when the sampling timestep γ is increased to 20, our model still leads in inference speed, outperforming GAS by more than twice. However, we also observed that as the sampling timestep continues to increase, the inference speed begins to slow down. This is because a higher number of sampling timestep directly increases the computational burden during the reverse generation process of the diffusion model, leading to longer inference times. Nevertheless, as a non-autoregressive generative model, the diffusion model can simultaneously infer all boundaries without needing to generate them sequentially from left to right. We set the sampling timestep to 5, achieving an excellent balance between efficiency and quality. Therefore, with a reasonable sampling timestep selection, our method still holds an advantage in inference speed. All inference times are averaged over five executions on a single GeForce RTX 3090.

4) Comparison With LLMs: We provide a detailed comparison between our model and large language models in terms of four dimensions: memory usage, the number of parameters, inference time, and F1 score, as shown in Table V. While large language models like GPT-3.5 and GPT-4 exhibit strong generative capabilities and versatility, they still have limitations in specific

TABLE V
COMPARISON WITH LLMS FOR DATASET REST14

Model	Memory	Num Params	Inf Time	F1(%)	Device
GPT-3.5	>80GB	175B	0.83s	49.48	Open API
GPT 4	>80GB	1760B	1.56s	35.37	Open API
Ours	5.39GB	0.08B	4.35s	73.74	3090 24GB

TABLE VI
ABLATION STUDY ON NOISE SCHEDULES

Noise Schedule	Timesteps	Lap14	Rest14	Rest15	Rest16
cosine	T=2000	84.92	87.13	**86.03**	86.51
	T=4000	85.11	86.72	83.56	85.31
	T=6000	85.40	**87.82**	83.76	**86.65**
	T=8000	**85.88**	87.36	83.72	85.88
linear	T=2000	**84.18**	87.04	83.45	86.16
	T=4000	83.90	87.16	**84.87**	85.70
	T=6000	83.86	87.12	84.16	**87.21**
	T=8000	83.31	**87.53**	84.53	86.31

tasks, particularly in terms of parameter size and memory usage, which significantly increase the demand for computational resources. Although our model has a slightly longer inference time, it consistently delivers excellent F1 scores, demonstrating high accuracy and reliability in specialized tasks. This comparison indicates that our approach offers superior performance with lower computational requirements in ASTE.

In the near future, we believe that fine-tuning large language models will play a crucial role in the ABSA domain. By fine-tuning these models for specific tasks, their performance in targeted applications can be further enhanced, leading to more specialized and efficient ABSA solutions.

C. Ablation Study

1) Noise Schedules: To compare the impact of different noise scheduling strategies on the diffusion process, we conduct ablation experiments on four datasets (Lap14, Rest14, Rest15, Rest16) using two noise injection policies: cosine and linear. Cosine is a noise scheduling strategy based on a cosine curve, where the noise level follows a cosine function over the diffusion steps. In contrast, linear is a noise scheduling strategy based on linear changes, where the noise level increases at a constant rate. The results are shown in Table VI, where the listed data represents the average F1 score for both ATE and OTE tasks. Experimental results indicate that the cosine noise scheduling strategy outperforms the linear strategy on Lap14, Rest14, and Rest15 datasets, with the most significant difference on the Lap14 dataset. On the average F1 score across timesteps, there is an improvement of 1.52% using cosine compared to linear. This indicates that the cosine strategy is more effective in improving model performance on these datasets. In addition, we also observe that the model performance shows fluctuations under changes in the noise timesteps. In the case of the Lap14 dataset, for example, the optimal performance is reached at a timestep of $t = 8000$ when using the cosine strategy, while the optimal performance is reached at a timestep of $t = 2000$ when using the linear strategy. This indicates that noise scheduling has a significant and important impact on model performance, and that choosing the appropriate noise strategy and timestep is crucial for optimizing the effectiveness of the diffusion process.

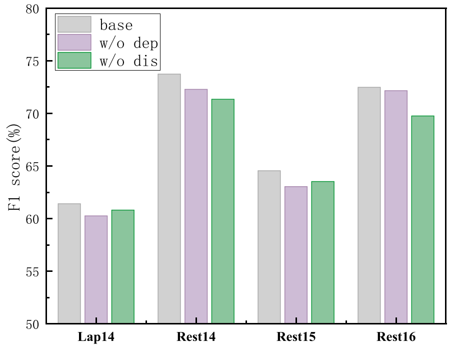

Fig. 6. Ablation study on noise schedules on pair representation.

2) Pair Representation: We also conduct ablation experiments to deeply explore the impact of distance representation and syntactic dependency in the span representation, with detailed results shown in Fig. 6. We observe that the performance of ASTE decreases significantly when distance embeddings are removed, especially on the two datasets Rest14 and Rest16. This suggests that introducing position embeddings for aspects and opinions in sentences is highly effective, as aspects and opinions in closer proximity are more likely to form valid aspect-opinion pairs. In addition, we find that performance also declines when syntactic dependency embeddings are removed. This further demonstrates that introducing syntactic information into the ASTE task is an effective strategy. In conclusion, the combination of distance embeddings and syntactic dependency embeddings significantly contributes to the model's performance in the ASTE task.

D. Performance in Complex Situations

In the triplets, many aspects and opinions consist of multiple words, as shown in Table I, where multi-word instances average more than half of the total. We conduct comparisons of our model with GTS and Span-ASTE under two different settings: 1) Single-word span: both aspect and opinion are single words; 2) Multi-word span: at least one of aspect and opinion is multi-word. We calculate the difference Δ_{F1} between the results of our method and the average results of GTS and Span-ASTE for a direct comparison. From the results in Table VII, our method demonstrates improvements in both single-word and multi-word settings, with particularly significant enhancement in the multi-word scenario. Moreover, whether it is GTS, Span-ASTE, or our DiffuSyn model, there is an observed improvement in performance for single-word settings and a significant decline in performance for multi-word scenarios, especially in the Rest14 dataset, where the performance drops by nearly 10%.

To further investigate the reasons for the performance decline in the case of multi-word, we conduct a detailed analysis focusing on multi-word aspects and opinions. The specific results are presented in Table IX. We observe that the results under the

TABLE VII
ANALYSIS OF SINGLE-WORD AND MULTI-WORD EVALUATION MODES ON THE ASTE TASK

Mode	Model	Lap14			Rest14			Rest15			Rest16		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
Single-Word	GTS	65.47	62.54	63.97	74.93	79.15	76.98	66.55	65.66	66.10	69.66	76.74	73.03
	Span-ASTE	68.09	65.98	67.02	79.12	79.60	79.36	70.23	70.71	70.47	71.66	77.91	74.65
	Ours	67.58	68.04	67.81	78.22	80.37	79.28	68.79	69.02	68.90	74.32	79.07	76.62
	Δ_{F1}	0.800	3.780	2.315	1.195	0.995	1.110	0.400	0.835	0.615	3.660	1.745	2.780
Multi-Word	GTS	52.26	41.27	46.12	56.85	49.26	52.78	50.28	47.34	48.77	56.63	55.29	55.95
	Span-ASTE	54.63	44.44	49.02	61.64	55.79	58.57	50.70	57.45	53.87	62.43	63.53	62.97
	Ours	59.61	48.02	53.19	66.78	57.86	62.00	56.99	58.51	57.74	67.11	60.00	63.35
	Δ_{F1}	6.165	5.165	5.620	7.535	5.335	6.325	6.500	6.115	6.420	7.580	0.590	3.890

TABLE VIII
CASE STUDY FOR ASTE

Review	BMRC	GAS	Ours
The portions of the food that came out were mediocre.	(food, mediocre, POS) ✗	(portions, mediocre, NEU) ✗ (food, mediocre, NEU) ✗	(The portions of the food, mediocre, NEU) ✓
My one complaint is that there was no internal CD drive.	(internal CD drive, complaint, NEG) ✓ (internal CD drive, no, NEG) ✓	✗ (internal CD drive, no, NEG) ✓	(internal CD drive, complaint, NEG) ✓ (internal CD drive, no, NEG) ✓
The food was extremely tasty, creatively presented and the wine excellent.	(food, tasty, POS) ✓ ✗ (wine, excellent, POS) ✓	(food, tasty, POS) ✓ ✗ (wine, excellent, POS) ✓	(food, tasty, POS) ✓ (food, creatively presented, POS) ✓ (wine, excellent, POS) ✓
Meal was very expensive for what you get.	(Meal, expensive, POS) ✗	(Meal, expensive, NEG) ✓	(Meal, expensive, NEG) ✓

TABLE IX
FURTHER COMPARISON WITH SPAN-ASTE ON THE MULTI-WORD ASPECT AND MULTI-WORD OPINION FOR ASTE

Dataset	Model	Multi-word Aspect			Multi-word Opinion			Δ_{F1}
		P	R	F1	P	R	F1	
Lap14	Span-ASTE	56.99	48.18	52.22	34.62	26.09	29.75	−22.47
	Ours	62.22	50.91	**56**	36.84	30.43	**33.33**	−22.67
Rest14	Span-ASTE	65.96	57.62	61.51	49.43	47.25	48.31	−13.20
	Ours	67.80	59.48	**63.37**	61.11	48.35	**53.99**	−9.38
Rest15	Span-ASTE	55.33	60.58	57.84	37.18	45.31	40.85	−16.99
	Ours	59.57	61.31	**60.43**	48.39	46.88	**47.62**	−12.81
Rest16	Span-ASTE	66.43	72.09	**69.16**	36.73	33.33	34.95	−34.21
	Ours	69.75	64.34	66.94	51.16	40.74	**45.36**	−21.58

multi-word opinion setting are significantly lower than those under the multi-word aspect setting, with an average F1 score decrease of 19.16%, which also indicates that the decline in performance for multi-word spans is primarily attributed to subpar performance in extracting multi-word opinions. This provides an entry point for future work emphasizing the accurate extraction of multi-word opinions.

E. Case Study

To further compare our model with other baselines, we select some sample cases from AST-Data-v2 for analysis. The predicted results are shown in Table VIII, where $\surd$ indicates that the prediction is consistent with the gold triplet, and ✗ indicates that the prediction is wrong. We choose BMRC and GAS as two baseline models for comparison. The experimental results show that our predictions are more consistent with the actual situation relative to the other two baselines, which further proves the effectiveness of our method.

In the first case, involving a multi-word aspect, both BMRC and GAS fail to identify the aspect boundaries, whereas our method accurately detects the boundaries. In the second and third cases, GAS and BMRC have the problem of missing opinions, while our method is able to detect all spans. In the last case, although BMRC successfully identifies the aspect and opinion, its sentiment polarity prediction was incorrect, whereas our method and GAS make successful predictions. These results highlight the outstanding performance of our method compared to other baselines when facing multi-word targets, multi-word opinions and multiple aspects and opinions in ASTE tasks.

VII. CONCLUSION

In this work, we propose a diffusion model-based framework with syntactic dependency (DiffuSyn) to identify the boundaries of aspects and opinions and solve ASTE tasks. We go through a step-by-step process of boundary denoising, and we successfully improve the aspect and opinion boundary accuracy, while speeding up the processing in a non-autoregressive manner. In addition, we introduce syntactic dependency of sentences when matching aspect and opinion and predicting sentiment polarity, so as to improve the accuracy of pairing. We evaluate the performance of our approach by conducting extensive experiments on a public benchmark including four datasets, and the results and analysis confirm the effectiveness of our work. In the future, we will focus on the case of multi-word opinion and explore more efficient methods to enhance the boundary recognition for this scenario.

VIII. Limitations

There are still some potential limitations or unconsidered issues with our method. First, our method requires learning the denoising process at a large timestep, which may lead to slow model convergence. Second, the syntactic analyses in our method may encounter errors in complex sentences, and these error propagations can affect downstream sentiment analysis tasks. We can design a more refined method to mine the syntactic dependencies of sentences to address complex dependency scenarios. Finally, like other current methods, our approach does not address the issue of multi-word opinions, which remains a direction requiring further research effort.

References

[1] W. Zhang, X. Li, Y. Deng, L. Bing, and W. Lam, "A survey on aspect-based sentiment analysis: Tasks, methods, and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 11, pp. 11019–11038, Nov. 2023.

[2] L. Zhu, Z. Zhu, C. Zhang, Y. Xu, and X. Kong, "Multimodal sentiment analysis based on fusion methods: A survey," *Inf. Fusion*, vol. 95, pp. 306–325, 2023.

[3] M. Hu, Y. Peng, Z. Huang, D. Li, and Y. Lv, "Open-domain targeted sentiment analysis via span-based extraction and classification," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 537–546.

[4] L. Xu, H. Li, W. Lu, and L. Bing, "Position-aware tagging for aspect sentiment triplet extraction," in *Proc. 2020 Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 2339–2349. [Online]. Available: https://aclanthology.org/2020.emnlp-main.183

[5] Y. Liu, Y. Zhou, Z. Li, J. Wang, W. Zhou, and S. Hu, "HIM: An end-to-end hierarchical interaction model for aspect sentiment triplet extraction," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 2272–2285, 2023.

[6] H. Peng, L. Xu, L. Bing, F. Huang, W. Lu, and L. Si, "Knowing what, how and why: A near complete solution for aspect-based sentiment analysis," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 8600–8607.

[7] Z. Wu, C. Ying, F. Zhao, Z. Fan, X. Dai, and R. Xia, "Grid tagging scheme for aspect-oriented fine-grained opinion extraction," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2020, pp. 2576–2585.

[8] P. Gupta, H. Schütze, and B. Andrassy, "Table filling multi-task recurrent neural network for joint entity and relation extraction," in *Proc. 26th Int. Conf. Comput. Linguistics*, 2016, pp. 2537–2547.

[9] H. Chen, Z. Zhai, F. Feng, R. Li, and X. Wang, "Enhanced multi-channel graph convolutional network for aspect sentiment triplet extraction," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 2974–2985.

[10] Y. Mao, Y. Shen, C. Yu, and L. Cai, "A joint training dual-MRC framework for aspect based sentiment analysis," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 13543–13551.

[11] S. Chen, Y. Wang, J. Liu, and Y. Wang, "Bidirectional machine reading comprehension for aspect sentiment triplet extraction," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 12666–12674.

[12] L. Xu, Y. K. Chia, and L. Bing, "Learning span-level interactions for aspect sentiment triplet extraction," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 4755–4766.

[13] Y. Chen, Z. Zhang, G. Zhou, X. Sun, and K. Chen, "Span-based dual-decoder framework for aspect sentiment triplet extraction," *Neurocomputing*, vol. 492, pp. 211–221, 2022.

[14] Y. Chen, C. Keming, X. Sun, and Z. Zhang, "A span-level bidirectional network for aspect sentiment triplet extraction," in *Proc. 2022 Conf. Empirical Methods Natural Lang. Process.*, 2022, pp. 4300–4309.

[15] H. Yan, J. Dai, T. Ji, X. Qiu, and Z. Zhang, "A unified generative framework for aspect-based sentiment analysis," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 2416–2429.

[16] W. Zhang, Y. Deng, X. Li, Y. Yuan, L. Bing, and W. Lam, "Aspect sentiment quad prediction as paraphrase generation," in *Proc. 2021 Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 9209–9219.

[17] W. Zhang, X. Li, Y. Deng, L. Bing, and W. Lam, "Towards generative aspect-based sentiment analysis," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 504–510.

[18] P. Liu, S. Joty, and H. Meng, "Fine-grained opinion mining with recurrent neural networks and word embeddings," in *Proc. 2015 Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1433–1443.

[19] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, "Recursive neural conditional random fields for aspect-based sentiment analysis," in *Proc. 2016 Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 616–626.

[20] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiaokui, "Coupled multi-layer attentions for co-extraction of aspect and opinion terms," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 3316–3322.

[21] A. B. P. Veyseh, N. Nouri, F. Dernoncourt, D. Dou, and T. H. Nguyen, "Introducing syntactic structures into target opinion word extraction with deep learning," in *Proc. 2020 Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 8947–8956.

[22] H. Zhao, L. Huang, R. Zhang, Q. Lu, and H. Xue, "SpanMlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 3239–3248. [Online]. Available: https://aclanthology.org/2020.acl-main.296

[23] S. Chen, J. Liu, Y. Wang, W. Zhang, and Z. Chi, "Synchronous double-channel recurrent network for aspect-opinion pair extraction," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6515–6524.

[24] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2256–2265.

[25] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 6840–6851.

[26] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=St1giarCHLP

[27] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 8780–8794.

[28] C. Saharia et al., "Palette: Image-to-image diffusion models," in *Proc. ACM SIGGRAPH Conf.*, 2022, pp. 1–10.

[29] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10684–10695.

[30] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "DiffWave: A versatile diffusion model for audio synthesis," in *Proc. Int. Conf. Learn. Representations*, 2021p. 17. [Online]. Available: https://openreview.net/forum?id=a-xFK8Ymz5J

[31] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "WaveGrad: Estimating gradients for waveform generation," in *Proc. 9th Int. Conf. Learn. Representations*, Austria, 2021. [Online]. Available: https://openreview.net/forum?id=NsMLjcFaO8O

[32] Q. Yi, X. Chen, C. Zhang, Z. Zhou, L. Zhu, and X. Kong, "Diffusion models in text generation: A survey," *PeerJ Comput. Sci.*, vol. 10, 2024, Art. no.e1905.

[33] X. Li, J. Thickstun, I. Gulrajani, P. S. Liang, and T. B. Hashimoto, "Diffusion-LM improves controllable text generation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 4328–4343.

[34] H. Yuan, Z. Yuan, C. Tan, F. Huang, and S. Huang, "Text diffusion with encoder-decoder transformers for sequence-to-sequence generation," in *Proc. 2024 Conf. North Amer. Chapter Ass. Comput. Linguistics: Human Lang. Technol.*, K. Duh, H. Gomez, and S. Bethard Eds., Mexico City, Mexico, Jun. 2024, pp. 22–39. [Online]. Available: https://aclanthology.org/2024.naacl-long.2/

[35] R. Strudel et al., "Self-conditioned embedding diffusion for text generation," 2022, *arXiv:2211.04236*.

[36] M. Reid, V. J. Hellendoorn, and G. Neubig, "DiffusER: Discrete diffusion via edit-based reconstruction," 2022, *arXiv:2210.16886*.

[37] Z. He et al., "DiffusionBERT: Improving generative masked language models with diffusion models," in *Proc. 61st Annu. Meet. Ass. Comput. Linguistics*, A. Rogers, J. Boyd-Graber, and N. Okazaki Eds., Toronto, Canada, Jul. 2023, pp. 4521–4534. [Online]. Available: https://aclanthology.org/2023.acl-long.248/

[38] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, "DiffusionNER: Boundary diffusion for named entity recognition," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics*, 2023, pp. 3875–3890. [Online]. Available: https://aclanthology.org/2023.acl-long.215

[39] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: Aspect based sentiment analysis," in *Proc. 8th Int. Workshop Semantic Eval.*, 2014, pp. 27–35. [Online]. Available: https://aclanthology.org/S14-2004

[40] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval-2015 task 12: Aspect based sentiment analysis," in *Proc. 9th Int. Workshop Semantic Eval.*, Jun. 2015, pp. 486–495. [Online]. Available: https://aclanthology.org/S15-2082

[41] M. Pontiki et al., "SemEval-2016 task 5: Aspect based sentiment analysis," in *Proc. 10th Int. Workshop Semantic Eval.*, 2016, pp. 19–30. [Online]. Available: https://aclanthology.org/S16-1002

[42] Z. Fan, Z. Wu, X.-Y. Dai, S. Huang, and J. Chen, "Target-oriented opinion words extraction with target-fused neural sequence labeling," in *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 2509–2518. [Online]. Available: https://aclanthology.org/N19-1259

[43] Z. Jin, M. Tao, X. Wu, and H. Zhang, "Span-based dependency-enhanced graph convolutional network for aspect sentiment triplet extraction," *Neurocomputing*, vol. 564, 2024, Art. no. 126966.

[44] Q. Sun, L. Yang, M. Ma, N. Ye, and Q. Gu, "MiniconGTS: A near ultimate minimalist contrastive grid tagging scheme for aspect sentiment triplet extraction," in *Proc. 2024 Conf. Emp. Methods Natural Lang. Process.*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen Eds., Miami, Florida, USA, Nov. 2024, pp. 2817–2834. [Online]. Available: https://aclanthology.org/2024.emnlp-main.165/

Qiuhua Yi received the B.Sc. degree in data science and Big Data technology from Jiaxing University, Jiaxing, China, in 2022. She is currently working toward the M.S. degree with the School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. Her research interests include social computing, natural language processing and affective computing.

Xiangjie Kong (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2004 and 2009 respectively. He was an Associate Professor with the School of Software, Dalian University of Technology, Dalian, China. He is currently a Professor with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou. He has authored or coauthored more than 200 scientific papers in international journals and conferences (with more than 180 indexed by ISI SCIE). His research interests include social computing, mobile computing, and natural language processing. He is a Distinguished Member of CCF and a Member of ACM.

Linan Zhu was born in Zhejiang Province of China, in 1982. He received the Doctor degree in mechanical manufacturing and automation from the Zhejiang University of Technology, Hangzhou, China, in 2014. He is currently an Associate Professor with the College of Computer Science and Technology, Zhejiang University of Technology. His research interests include natural language processing, sentiment analysis, opinion mining, and machine learning. He is also a member of China Computer Federation.

Chenwei Zhang received the Ph.D. degree in information science from Indiana University, Bloomington, IN, USA, with a Ph.D. minor in computer science. She is currently an Assistant Professor with the Faculty of Education, University of Hong Kong, Hong Kong. Her research interests include science of science, which studies the mechanisms underlying the doing of science and data science, such as educational data mining, where a variety of computational approaches including machine learning and deep learning are applied for knowledge discovery.

Guojiang Shen received the B.Sc. degree in control theory and control engineering and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 1999 and 2004, respectively. He is currently a Professor with the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou. His research interests include artificial intelligence, Big Data analytics, and intelligent transportation systems.